



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:)	Art Unit: 2111
)	
Steven L. CRETA <i>et al.</i>)	Examiner: Auve, Glenn Allen.
)	
Serial No: 09/940,292)	
)	
Filed: August 27, 2001)	
)	
For: <i>MECHANISM FOR PRESERVING</i>)	
<i>PRODUCER-CONSUMER ORDERING ACROSS</i>)	
<i>AN UNORDERED INTERFACE</i>)	
)	

DECLARATION OF ROGER R. WISE UNDER 37 CFR 1.131

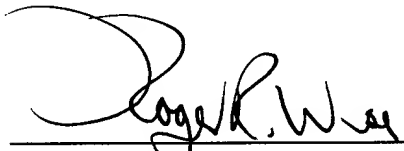
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

I, Roger R. Wise, declare and state as follows:

1. I am the patent attorney who supervised the preparation of, and filed the, the above-referenced patent application. Eric S. Chen, an associate attorney in my office, prepared the application.
2. On or about August 23, 2001, Mr. Chen prepared, in the United States, a draft of the application. After my review of the draft, Mr. Chen sent the draft as an attachment to an email (on August 23, 2001) to Kenneth Creta, one of the inventors, and carbon copied Brandon Congdon, Tony Rand, and Deepak Ramachandran, the other inventors, for their review. All the inventors were in the United States. I was carbon copied on that email as well. Attached as "Exhibit A" is a true and accurate copy of the email and the attached draft of the application.

3. The draft of the application, i.e., the specification and the drawings, show conception, in the United States, of the invention claimed in U.S. Patent Application Serial No. 09/940,292. The draft application is substantially identical to the application as filed.
4. The application was constructively reduced to practice four days after the draft was sent to the inventors. Thus the date of invention is at least as early as August 23, 2001.
5. I hereby declare that all statements made of my own knowledge are true and that all statements made on information and belief are believed to be true, and, further, that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application and any patent issued thereon.



Roger R. Wise

May 11, 2004

Date

**APPLICATION FOR
UNITED STATES PATENT
IN THE NAME OF**

**KENNETH C. CRETA; BRADFORD B. CONGDON; TONY RAND;
AND DEEPAK RAMACHANDRAN**

FOR

**MECHANISM FOR PRESERVING PRODUCER-CONSUMER ORDERING ACROSS
AN UNORDERED INTERFACE**

Prepared By:

**PILLSBURY WINTHROP LLP
725 South Figueroa Street, Suite 2800
Los Angeles, CA 90017-5406
Telephone (213) 488-7100
Facsimile (213) 629-1033**

Attorney Docket No: 81674-276924

Client Docket No.: P-12778

Express Mail No.:

TITLE OF THE INVENTION

MECHANISM FOR PRESERVING PRODUCER-CONSUMER ORDERING ACROSS AN
UNORDERED INTERFACE

5 BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention generally relates to an input/output (I/O) hub. More particularly,
the present invention relates to an I/O hub that is adapted to implement Producer-Consumer
(P/C) ordering rules across an interface that is inherently unordered in a multi-processor
10 computer system architecture.

2. Discussion of the Related Art

Multi-processor computer systems are designed to accommodate a number of central
processing units (CPUs), coupled via a common system bus or switch to a memory and a number
15 of external input/output devices. The purpose of providing multiple central processing units is to
increase the performance of operations by sharing tasks between the processors. Such an
arrangement allows the computer to simultaneously support a number of different applications
while supporting I/O components that are, for example, communicating over a network and
displaying images on attached display devices. Multi-processor computer systems are typically
20 utilized for enterprise and network server systems.

An input/output hub may be provided as a connection point between various input/output
bridge components, to which input/output components are attached, and ultimately to the central
processing units. Many input/output components are Peripheral Component Interconnect (PCI)

(“PCI Local Bus Specification, Revision 2.1, June 1, 1995, from the PCI Special Interest Group (PCI-SIG)) devices and software drivers that adhere to the PCI Producer-Consumer (P/C) model and its ordering rules and requirements. (“PCI Local Bus Specification”, Revision 2.1, Appendix E, “System Transaction Ordering”.) For example, these ordering rules allow writes to
5 be posted for higher performance while ensuring “correctness”. Posting means that the transaction is captured by an intermediate agent, e.g., a bridge from one bus to another, so that the transaction completes at the source before it actually completes at the intended destination. Posting allows the source to proceed with the next operation while the transaction is still making its way through the system to its ultimate destination. In other words, write posting in a PCI
10 device means that the writes that are issued are not expected to return a “complete” response. That is, when posted writes are issued, there is no confirmation returned indicating that the write is completed. The term “correctness” implies that a flag or semaphore may be utilized to guard a data buffer between a Producer-Consumer pair.

Coherent interfaces interconnecting the I/O hub and, ultimately, to the processors, are
15 inherently unordered. Therefore, ordering rules under the P/C model are more restrictive than those for a coherent interface, which may have no ordering rules at all. Coherent interfaces, such as a front-side bus or an Intel Scalability Port, are inherently ordered because the processors for which the coherent interface was designed for are complex devices. These processors have the intelligence to distinguish when ordering is required and when it is not. Therefore, in general,
20 coherent interfaces can treat completions independently of requests (in either direction). PCI devices, however, are generally not this complex and are more cost-sensitive, and therefore rely on the system ordering rules to avoid deadlocks. PCI ordering rules do allow some flexibility in relaxing the ordering requirements of specific transactions, though.

It is particularly beneficial to retain the use of PCI devices and devices that follow the P/C ordering model, as they are generally designed toward cost-sensitivity. Accordingly, what is needed is a cost-effective optimized chipset implementation that bridges an ordered domain (one which requires PCI ordering and follows the P/C ordering model) and an unordered domain, such as a coherent interface in connection with a plurality of processor units, without any additional software or hardware intervention. Because a PCI device is generally designed towards cost-sensitivity and may not exploit the relaxations in the PCI ordering rules, there is a need for a system that can exploit the performance optimizations allowed with the PCI ordering rules by employing all of the ordering relaxation capabilities on behalf of these devices, while at the same time avoiding any deadlock vulnerabilities and performance penalties.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 illustrates an input/output hub according to an embodiment of the present invention;

Fig. 2A illustrates an inbound transaction through an inbound ordering queue (IOQ) according to an embodiment of the present invention;

Fig. 2B illustrates an outbound transaction through an outbound ordering queue (OOQ) according to an embodiment of the present invention; and

Fig. 3 illustrates an input/output system architecture according to an embodiment of the present invention.

DETAILED DESCRIPTION

Fig. 1 illustrates an input/output hub according to an embodiment of the present invention. The I/O hub 100 includes an ordered domain and an unordered domain. Within the ordered domain, one or more transaction queues 102, 104 facilitate the inbound and outbound transactions between the I/O component(s) 160, 170 and the unordered protocol 110. Each transaction queue 102, 104 includes an inbound ordering queue (IOQ) 120, an IOQ read bypass buffer (RBB) 125, an outbound ordering queue (OOQ) 130, and an OOQ read bypass buffer (RBB) 135.

Within the unordered domain, an inbound multiplexer 180 receives data and signals from the transaction queue(s) 102, 104 of the ordered domain (and more specifically, from the IOQ 120 and the IOQ RBB 125). An outbound demultiplexer 190 within the unordered domain receives data and signals from the unordered protocol 110, such as a coherent interface like the Scalability Port, for transmission to the ordered domain (and more specifically, to the OOQ 130 of the transaction queue(s) 102, 104).

At least one P/C ordered input/output interface 140, 150 is provided to connect with the input/output devices or components 160, 170, such as PCI devices. The P/C ordered interface 140, 150 typically does not directly connect with the I/O devices or components 160, 170, though. An intermediary device, such as a hub-link or input/output bridge, like an Intel P64H2 Hub Interface-to-PCI Bridge, or a VXB InfiniBand ("InfiniBand Architecture Specification", version 1.0, June 19, 2001, from the InfiniBand Trade Association) Bridge, is generally connected to the P/C ordered interface 140, 150, to which the I/O devices or components 160, 170 connect. Each P64H2 bridge, for example, has two PCI-X ("PCI-X Specification", Revision 1.0a, August 29, 2000, from the PCI-SIG) segments to which I/O devices or components 160,

170 may connect. PCI-X is a high-performance extension to the PCI local bus having increased bandwidth and bus performance.

The I/O hub 100 according to an embodiment of the present invention is “cut” into two domains: an ordered domain and an unordered domain. The ordered domain adheres to the
5 Producer-Consumer ordering rules described in the PCI specification and may be designed in many different ways. The unordered domain has no ordering rules. By implementing the I/O hub 100 according to the layered approach of the present invention, Producer-Consumer ordering across an unordered interface may be preserved.

Inbound ordering queues (IOQs) 120 are responsible for enqueueing inbound read and
10 write transactions/requests targeting the main memory or a peer I/O component. The IOQ 120 is preferably configured in a first-in-first-out (FIFO) manner enforcing that inbound read and write transactions/requests are not allowed to bypass inbound writes (i.e., write data). Moreover, outbound read and write completions (data returning for reads targeting an I/O component) are also enqueued in the IOQ 120, along with any other outbound special cycles. Utilizing this
15 configuration, Producer-Consumer “correctness” may be ensured.

Under the PCI ordering rules, posted writes are permitted. However, in the unordered domain, posted write transactions are not allowed. Accordingly, both read and write transactions require a transaction completion. Therefore, writes in the IOQ 120 are issued to the unordered domain and are not deallocated until the unordered interface returns a completion (to the OQ
20 130).

When a peer-to-peer transaction is issued, it is not permitted to the destination interface (either on the same I/O hub or a different I/O hub) until after all prior writes in the IOQ 120 have been completed. This restriction ensures proper ordering when the data and semaphore are

located in different destinations, e.g., the first write is data to the main memory and the peer-to-peer write is for a semaphore on the peer I/O component.

With respect to peer-to-peer write transactions that flow between two I/O hubs, there is some time where the posted write flows through the unordered fabric before reaching the ordered domain in the destination I/O hub. Therefore, the write (even though it is peer-to-peer and targets the ordered domain) must not allow subsequent accesses to proceed until the peer write is guaranteed to be in the ordered domain of the destination. This requirement ensures “completion” for the posted write.

The number of IOQs 120 implemented depends on the number of independent data streams for which the I/O hub is optimized. At a minimum, one queue will provide correct behavior, but one queue per stream would relax the ordering constraints between independent data streams.

The outbound ordering queues (OOQs) 130, along with the OOQ read bypass buffer (RBB) 135, maintain Producer-Consumer ordering by holding both outbound transactions (e.g., read and write requests) as well as completions for inbound transactions. As stated before, according to an embodiment of the present invention, the unordered domain requires completions even for write transactions. The I/O hub 100 is responsible for posting these writes for optimal performance in the ordered domain and does so by issuing a completion response (from the OOQ 130) for the write only after it has reached the OOQ 130. Similarly, reads could theoretically fill up the OOQ 130. In order to prevent this “back pressure” from flowing into the unordered domain (which could prevent write forward-progress), reads are retried at the ordered domain boundary line when permissible.

The IOQ 120 and the OQ 130 each have at least one corresponding read bypass buffer (RBB) 125, 135, respectively. The read bypass buffers 125, 135 allow posted writes and read completions to make progress past stalled read requests waiting for their completions to return. They apply to both inbound and outbound traffic. That is, when a posted write or read completion needs to progress through the IOQ 120 or OQ 130, the immediate contents (usually stalled read transactions/requests) within the IOQ 120 or OQ 130 are “pushed” aside into the respective RBBs 125, 135 so as to allow the posted write or read completion to progress through the IOQ 120 or OQ 130. Then, the first “pushed aside” task in the queue of the RBB 125, 135 is attempted when the blocking condition causing the stall no longer exists. The contents within the RBBs 125, 135 and subsequent content within the IOQ 120 and OQ 130 are then arbitrated to be completed. The read bypass buffers 125, 135 ensure deadlock free operation in the ordered domain.

According to an embodiment of the present invention, a transaction queue 102, 104 (which has an IOQ 120 and an OQ 130) is provided with each P/C ordered interface 140, 150. Although the embodiment illustrated in Fig. 1 shows two transaction queues 102, 104 and a corresponding P/C ordered interface 140, 150 for each transaction queue 102, 104, any suitable configuration and numbers of transaction queues and P/C ordered interfaces may be utilized.

Fig. 2A illustrates an inbound transaction through an inbound ordering queue (IOQ) according to an embodiment of the present invention. The P/C ordered interface 140, 150 (by direction of the I/O component 160, 170) issues 202 a read or write transaction/request to the IOQ 120 of the I/O hub 100. The read or write transaction is enqueued 204 in the IOQ 120. If one or more transactions are stalled in the IOQ 120, the contents of the IOQ 120 are pushed 206 aside into the IOQ read bypass buffer 125 so as to permit inbound write transaction(s) or write

completion(s) to progress through the IOQ 120 and to the unordered protocol 110. Otherwise, the read or write transactions enqueued in the IOQ 120 are forwarded 208 to the unordered protocol 110, preferably in a first-in-first-out (FIFO) fashion.

Fig. 2B illustrates an outbound transaction through an outbound ordering queue (OOQ) according to an embodiment of the present invention. At least one of a read or write transaction/request and a read completion are issued 220 from the unordered protocol 110, such as a coherent interface like a Scalability Port, to the OOQ 130 of the I/O hub 100. The at least one of the read or write transaction and the read completion are enqueued 222 in the OOQ 130. The OOQ 130 forwards 224 the at least one of the read or write transaction and the read completion from the OOQ 130 to the P/C ordered interface 140, 150, and ultimately to the I/O component 160, 170. When a write transaction is issued from the OOQ 130, it is then removed 226 from the OOQ 130. When a read transaction is issued from the OOQ 130, it is also then removed 228 from the OOQ 130. Removal of the read or write transactions from the OOQ 130 once they have been issued from the OOQ 130 ensures that each transaction has a completion.

Fig. 3 illustrates an input/output system architecture according to an embodiment of the present invention. As discussed above, the I/O hub 100 may include P/C ordered interfaces that are coupled to an intermediary device, such as a hub-link or input/output bridge, like a PCI-X bridge 360 or an InfiniBand bridge 370. The I/O components or devices 160, 170 (of Fig. 1) then connect to the intermediary devices 360, 370. The I/O hub 100 may also include an I/O interface that connects to a legacy input/output bridge 350 to handle connections with legacy I/O components or devices.

The I/O hub 100 is adapted to connect to a coherent interface, such as a Scalability Port 340, which is a cache-coherent interface optimized for scalable multi-node systems that maintain

coherency between all processors and their caches. The Scalability Port 340 in turn may connect to at least one Scalability Node Controller 320, which controls the interface between the processors 310, the main memory 330, e.g., dynamic random access memory (DRAM), and the Scalability Port 340.

5 In summary, the I/O hub 100 according to the present invention permits retention of the use of PCI devices and devices that follow the P/C ordering model, which are generally designed towards cost-sensitivity. The I/O hub 100 provides a cost-effective optimized chipset implementation, such as in the Intel 870 chipset, that bridges an ordered domain (one which requires PCI ordering and follows the P/C ordering model) and an unordered domain, such as a
10 coherent interface, without any additional software or hardware intervention. Because a PCI device is generally designed towards cost-sensitivity and may not exploit the relaxations in the PCI ordering rules, the I/O hub 100 of the present invention exploits the performance optimizations allowed with the PCI ordering rules by employing all of the ordering relaxation capabilities on behalf of these devices, while at the same time avoiding any deadlock
15 vulnerabilities and performance penalties.

 While the description above refers to particular embodiments of the present invention, it will be understood that many modifications may be made without departing from the spirit thereof. The accompanying claims are intended to cover such modifications as would fall within the true scope and spirit of the present invention. The presently disclosed embodiments are
20 therefore to be considered in all respects as illustrative and not restrictive, the scope of the invention being indicated by the appended claims, rather than the foregoing description, and all changes that come within the meaning and range of equivalency of the claims are therefore intended to be embraced therein.

WHAT IS CLAIMED IS:

1 1. An input/output hub, comprising:
2 an inbound ordering queue (IOQ) to receive inbound transactions, wherein all
3 read and write transactions have a transaction completion, peer-to-peer transactions are
4 not permitted to reach a destination until after all prior writes in the IOQ have been
5 completed, and a write in a peer-to-peer transaction does not permit subsequent accesses
6 to proceed until the write is guaranteed to be in an ordered domain of the destination;
7 an IOQ read bypass buffer to receive contents pushed from the IOQ to permit
8 posted writes and read completions to progress through the IOQ;
9 an outbound ordering queue (OOQ) to store outbound transactions and
10 completions of the inbound transactions, and to issue a write completion for a posted
11 write;
12 an OOQ read bypass buffer to receive contents pushed from the OOQ
13 to permit the posted writes and the read completions to progress through the OOQ; and
14 an unordered domain to receive the inbound transactions transmitted from the
15 IOQ and to receive the outbound transactions transmitted from an unordered protocol.

1 2. The input/output hub according to claim 1, wherein the IOQ does not permit the
2 inbound read and write transactions to bypass inbound write data.

1 3. The input/output hub according to claim 1, wherein the unordered protocol is a
2 coherence interface.

1 4. The input/output hub according to claim 3, wherein the coherent interface is a
2 Scalability Port.

1 5. An input/output hub, comprising:
2 an ordered domain, including:
3 an inbound ordering queue (IOQ) to receive and transmit inbound
4 transactions, wherein inbound read and write transactions are not permitted to
5 bypass inbound write data, all the read and write transactions have a transaction
6 completion, peer-to-peer transactions are not permitted to reach a destination until
7 after all prior writes in the IOQ have been completed, and a write in a peer-to-peer
8 transaction does not permit subsequent accesses to proceed until the write is
9 guaranteed to be in an ordered domain of the destination,
10 an IOQ read bypass buffer to receive contents pushed from the IOQ to
11 permit posted writes and read completions to progress through the IOQ,
12 an outbound ordering queue (OOQ) to store outbound transactions and
13 completions of the inbound transactions, and to issue a write completion for a
14 posted write, and
15 an OOQ read bypass buffer to receive contents pushed from the OOQ
16 to permit the posted writes and the read completions to progress through the
17 OOQ; and
18 an unordered domain, in communication with an unordered protocol, including:
19 an inbound multiplexer to receive the inbound transactions from the
20 ordered domain to the unordered protocol, and

21 an outbound demultiplexer to receive the outbound transactions from the
22 unordered protocol to the ordered domain.

1 6. The input/output hub according to claim 5, further including at least one
2 Producer-Consumer ordered interface in communication with the ordered domain.

1 7. The input/output hub according to claim 6, further including an input/output
2 device connected with the Producer-Consumer ordered interface.

1 8. The input/output hub according to claim 7, further including an intermediary
2 device interconnecting the Producer-Consumer ordered interface and an input/output device.

1 9. The input/output hub according to claim 7, wherein the input/output device is a
2 Peripheral Component Interconnect (PCI) device.

1 10. The input/output hub according to claim 5, wherein the unordered protocol is a
2 coherence interface.

1 11. The input/output hub according to claim 10, wherein the coherent interface is a
2 Scalability Port.

1 12. An input/output system, comprising:
2 an ordered domain, including:

3 an inbound ordering queue (IOQ) to receive and transmit inbound
4 transactions, wherein inbound read and write transactions are not permitted to
5 bypass inbound write data, all the read and write transactions have a transaction
6 completion, peer-to-peer transactions are not permitted to reach a destination until
7 after all prior writes in the IOQ have been completed, and a write in a peer-to-peer
8 transaction does not permit subsequent accesses to proceed until the write is
9 guaranteed to be in an ordered domain of the destination,

10 an IOQ read bypass buffer to receive contents pushed from the IOQ to
11 permit posted writes and read completions to progress through the IOQ,

12 an outbound ordering queue (OOQ) to store outbound transactions and
13 completions of the inbound transactions, and to issue a write completion for a
14 posted write,

15 an OOO read bypass buffer to receive contents pushed from the OOO
16 to permit the posted writes and the read completions to progress through the
17 OOO;

18 an unordered domain, in communication with an unordered protocol, including:

19 an inbound multiplexer to receive the inbound transactions from the
20 ordered domain to the unordered protocol, and

21 an outbound demultiplexer to receive the outbound transactions from the
22 unordered protocol to the ordered domain;

23 a Producer-Consumer ordered interface in communication with the ordered
24 domain;

25 an input/output device connected with the Producer-Consumer ordered
26 interface; and
27 a coherent interface within the unordered protocol in communication with the
28 unordered domain.

1 13. The input/output system according to claim 12, wherein the coherent interface is a
2 Scalability Port.

1 14. The input/output system according to claim 12, wherein the input/output device is
2 a Peripheral Component Interconnect (PCI) device.

1 15. The input/output system according to claim 12, further including an intermediary
2 device interconnecting the Producer-Consumer ordered interface and the input/output device.

1 16. An input/output system, comprising:
2 an ordered domain having a first transaction queue and a second transaction
3 queue, wherein the first transaction queue and the second transaction queue each include:
4 an inbound ordering queue (IOQ) to receive inbound transactions, wherein
5 inbound read and write transactions are not permitted to bypass inbound write
6 data, all the read and write transactions have a transaction completion, peer-to-
7 peer transactions are not permitted to reach a destination until after all prior writes
8 in the IOQ have been completed, and a write in a peer-to-peer transaction does

9 not permit subsequent accesses to proceed until the write is guaranteed to be in an
10 ordered domain of the destination,

11 an IOQ read bypass buffer to receive contents pushed from the IOQ to
12 permit posted writes and read completions to progress through the IOQ,

13 an outbound ordering queue (OOQ) to store outbound transactions and
14 completions of the inbound transactions, and to issue a write completion for a
15 posted write,

16 an OOQ read bypass buffer to receive contents pushed from the OOQ
17 to permit the posted writes and the read completions to progress through the
18 OOQ;

19 an unordered domain, in communication with an unordered protocol, including:

20 an inbound multiplexer to receive the inbound transactions from the
21 ordered domain to the unordered protocol, and

22 an outbound demultiplexer to receive the outbound transactions from the
23 unordered protocol to the ordered domain;

24 a first Producer-Consumer ordered interface in communication with the first
25 transaction queue;

26 a first input/output device connected with the first Producer-Consumer ordered
27 interface;

28 a second Producer-Consumer ordered interface in communication with the second
29 transaction queue;

30 a second input/output device connected with the second Producer-Consumer
31 ordered interface; and

32 a coherent interface within the unordered protocol in communication with the
33 unordered domain.

1 17. The input/output system according to claim 16, wherein the coherent interface is a
2 Scalability Port

1 18. The input/output system according to claim 16, wherein the first input/output
2 device is a Peripheral Component Interconnect (PCI) device.

1 19. The input/output system according to claim 16, wherein the second input/output
2 device is a Peripheral Component Interconnect (PCI) device.

1 20. The input/output system according to claim 16, further including a first
2 intermediary device interconnecting the first Producer-Consumer ordered interface and the first
3 input/output device.

1 21. The input/output system according to claim 16, further including a second
2 intermediary device interconnecting the second Producer-Consumer ordered interface and the
3 second input/output device.

1 22. A computer system, comprising:
2 a plurality of processor units having access to caches;
3 a main memory;

4 a coherent interface to maintain coherency between the processor units and their
5 caches;

6 a scalability node controller interconnecting the processor units, the main
7 memory, and the coherent interface to control interface therebetween; and

8 an input/output hub in communication with the coherent interface, including:

9 an inbound ordering queue (IOQ) to receive inbound transactions, wherein
10 all read and write transactions have a transaction completion, peer-to-peer
11 transactions are not permitted to reach a destination until after all prior writes in
12 the IOQ have been completed, and a write in a peer-to-peer transaction does not
13 permit subsequent accesses to proceed until the write is guaranteed to be in an
14 ordered domain of the destination;

15 an IOQ read bypass buffer to receive contents pushed from the IOQ to
16 permit posted writes and read completions to progress through the IOQ;

17 an outbound ordering queue (OOQ) to store outbound transactions and
18 completions of the inbound transactions, and to issue a write completion for a
19 posted write;

20 an OOQ read bypass buffer to receive contents pushed from the OOQ to
21 permit the posted writes and the read completions to progress through the OOQ;

22 and

23 an unordered domain to receive the inbound transactions transmitted from
24 the IOQ and to receive the outbound transactions from the coherent interface.

1 23. The computer system according to claim 22, wherein the IOQ does not permit the
2 inbound read and write transactions to bypass inbound write data.

1 24. The computer system according to claim 22, wherein the coherence interface is an
2 unordered protocol.

1 25. The computer system according to claim 22, wherein the coherent interface is a
2 Scalability Port.

ABSTRACT OF THE DISCLOSURE

An input/output hub includes an inbound ordering queue (IOQ) to receive inbound transactions. All read and write transactions have a transaction completion. Peer-to-peer transactions are not permitted to reach a destination until after all prior writes in the IOQ have been completed. A write in a peer-to-peer transaction does not permit subsequent accesses to proceed until the write is guaranteed to be in an ordered domain of the destination. An IOQ read bypass buffer is provided to receive contents pushed from the IOQ to permit posted writes and read completions to progress through the IOQ. An outbound ordering queue (OOQ) stores outbound transactions and completions of the inbound transactions. The OOQ also issues write completions for posted writes. An OOQ read bypass buffer is provided to receive contents pushed from the OOQ to permit posted writes and read completions to progress through the OOQ. An unordered domain within the input/output hub receives the inbound transactions transmitted from the IOQ and receives the outbound transactions transmitted from an unordered protocol.

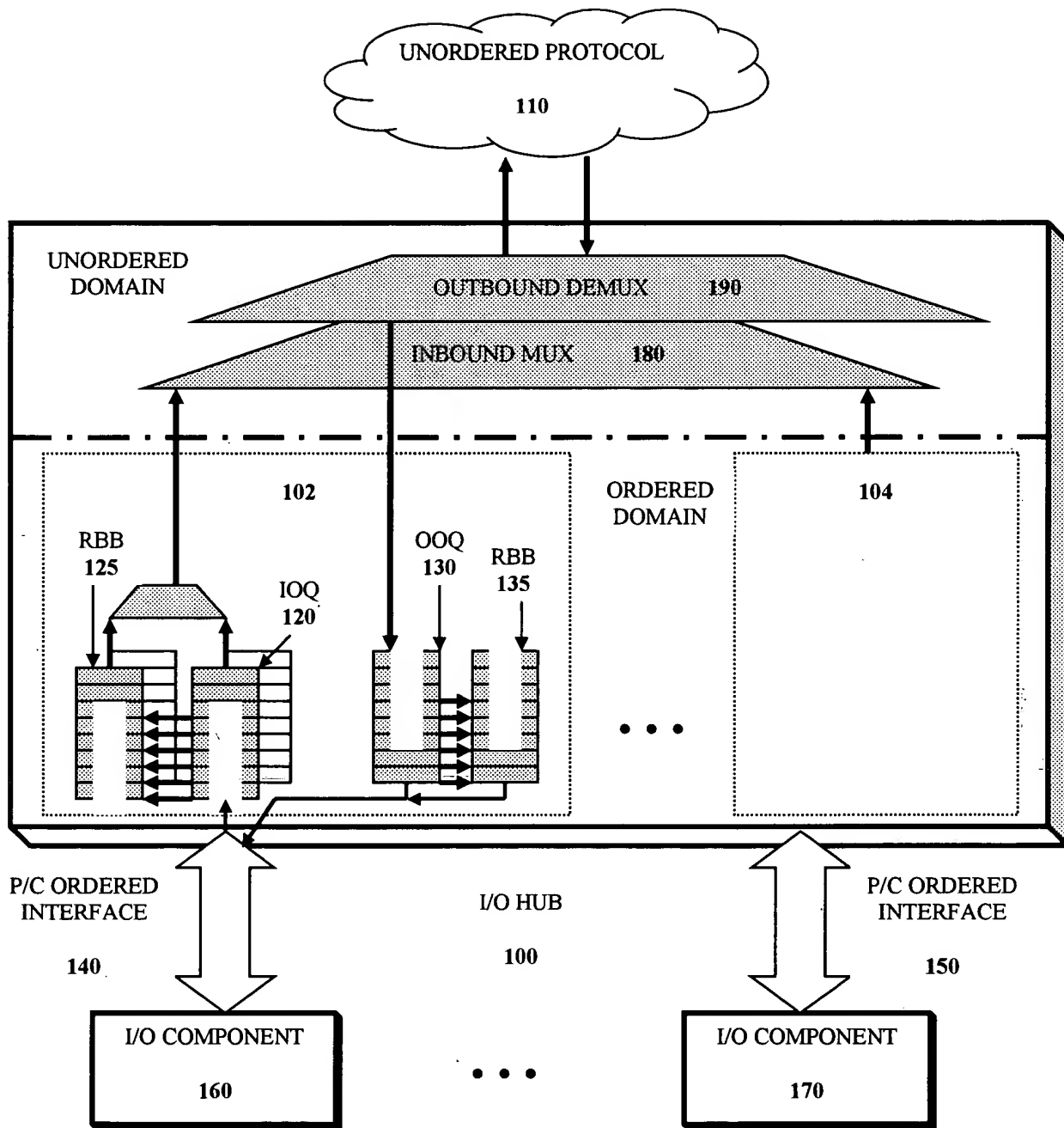


FIG. 1

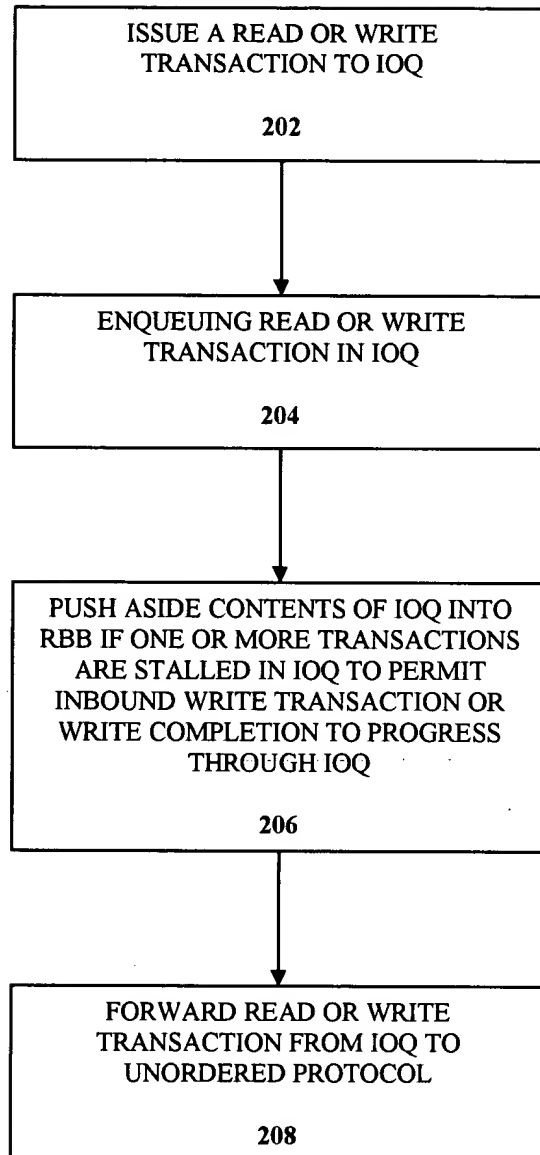


FIG. 2A

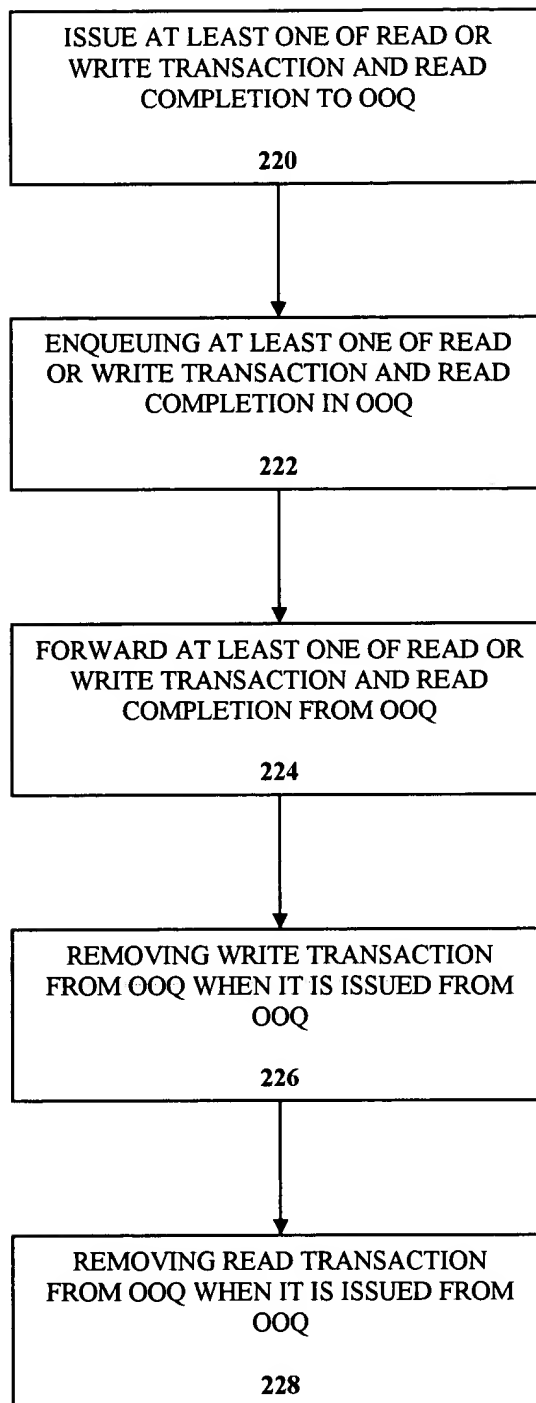


FIG. 2B

300

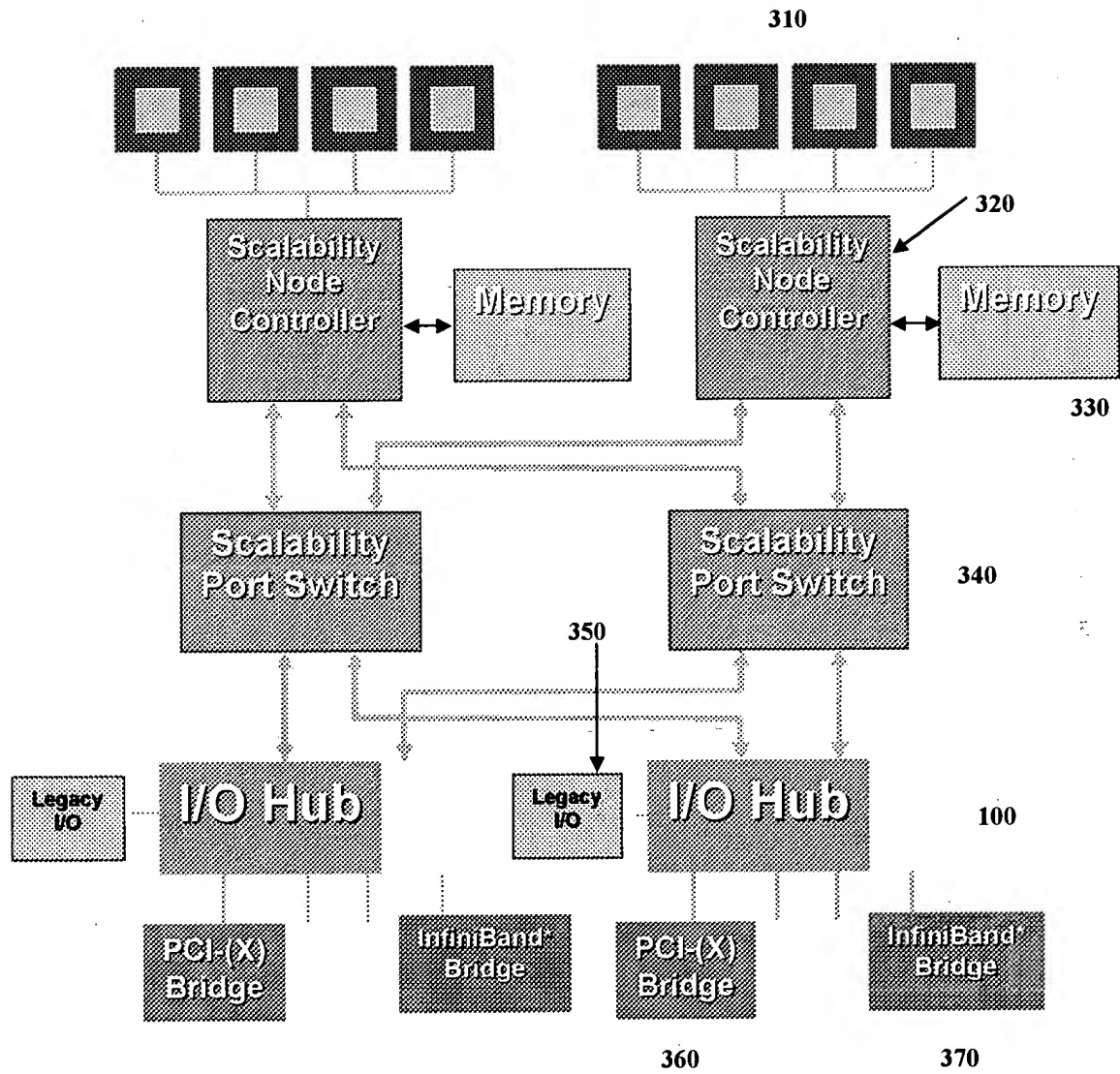


FIG. 3